

Prior-Independent Mechanisms for Scheduling

Shuchi Chawla* Jason D. Hartline† David Malec‡ Balasubramanian Sivan§

Abstract

We study the makespan minimization problem with unrelated selfish machines under the assumption that job sizes are stochastic. We design simple truthful mechanisms that under various distributional assumptions provide constant and sublogarithmic approximations to expected makespan. Our mechanisms are prior-independent in that they do not rely on knowledge of the job size distributions. Prior-independent approximation mechanisms have been previously studied for the objective of revenue maximization [13, 11, 26]. In contrast to our results, in prior-free settings no truthful anonymous deterministic mechanism for the makespan objective can provide a sublinear approximation [3].

*Computer Sciences Dept., University of Wisconsin - Madison. shuchi@cs.wisc.edu.

†EECS, Northwestern University. hartline@eecs.northwestern.edu.

‡Computer Sciences Dept., University of Wisconsin - Madison. dmalec@cs.wisc.edu.

§Computer Sciences Dept., University of Wisconsin - Madison. balu2901@cs.wisc.edu.

1 Introduction

We study the problem of scheduling jobs on machines to minimize makespan in a strategic context. The makespan is the longest it takes any of the machines to complete the work assigned by the schedule. The running time or size of a job on a machine is drawn from a fixed distribution, and is a private input known to the machine but not to the optimizer. The machines are unrelated in the sense that the running time of a job on distinct machines may be distinct. A scheduling mechanism solicits job running times from the machines and determines a schedule as well as compensation for each of the machines. The machines are strategic and try to maximize the compensation they receive minus the work they perform. We are interested in understanding and quantifying the loss in performance due to the strategic incentives of the machines who may misreport the job running times.

A primary concern in the theory of mechanism design is to understand the compatibility of various objectives of the designer with the incentives of the participants. As an example, maximizing social welfare is incentive compatible; the Vickrey-Clarke-Groves (VCG) mechanism obtains this socially optimal outcome in equilibrium [27, 9, 15]. For most other objectives, however, the optimal solution ignoring incentives (a.k.a. the *first-best* solution) cannot be implemented in an incentive compatible manner. This includes, for example, the objectives of revenue maximization, welfare maximization with budgets, and makespan minimization with unrelated machines. For these objectives there is no incentive compatible mechanism that is best on every input. The classical economic approach to mechanism design thus considers inputs drawn from a distribution (a.k.a. the *prior*) and looks for the mechanism that maximizes the objective in expectation over the distribution (a.k.a. the *second-best* solution).

The second-best solution is generally complex and, by definition, tailored to specific knowledge that the designer has on the distribution over the private information (i.e., the input) of the agents. The non-pointwise optimality, complexity, and distributional dependence of the second-best solution motivates a number of mechanism design and analysis questions.

price of anarchy: For any distribution over inputs, bound the gap between the first-best (optimal without incentives) and second-best (optimal with incentives) solutions (each in expectation over the input).

computational tractability: For any distribution over inputs, give a computationally tractable implementation of the second-best solution, or if the problem is intractable give a computationally tractable approximation mechanism.

simplicity: For any distribution over inputs, give a simple, practical mechanism that approximates the second-best solution.

prior independence: Give a single mechanism that, for all distributions over inputs, approximates the second-best solution.

These questions are inter-related. As the second-best mechanism is often complex, the price of anarchy can be bounded via a lower bound on the second-best mechanism as given by a simple approximation mechanism. Similarly, to show that a mechanism is a good approximation to second-best the upper bound given by the first-best solution can be used. Importantly though, if the first-best solution does not permit good approximation mechanisms then a better bound on the second-best solution should be sought. Each of the questions above can be further refined by consideration with respect to a large class of priors (e.g. identical distributions).

The prior-independence question gives a middle ground between worst-case mechanism design and Bayesian mechanism design. It attempts to achieve the best of both worlds in the tradeoff between informational efficiency and approximate optimality. Its minimal usage of information about the setting makes it robust. A typical side-effect of this robustness is simple and natural mechanisms; indeed, our prior-independent mechanisms will be simple, computationally tractable, and also enable a bound on the price of anarchy.

The literature on prior-independent mechanism design has focused primarily on the objective of revenue maximization. Hartline and Roughgarden [17] show that with sufficient competition, the welfare maximizing (VCG) mechanism also attains good revenue. This result enables the prior-independent approximation mechanism for single-item auctions of Dhangwatnotai, Roughgarden, and Yan [13] and the multi-item approximation mechanisms of Devanur et al. [11] and Roughgarden et al. [26]. Importantly, in single-item auctions the agents' private information is single-dimensional whereas in multi-item auctions it is multi-dimensional. There are several interesting and challenging directions in prior-independent mechanism design: (1) non-linear objectives, (2) general multi-parameter preferences of agents, (3) non-downwards-closed feasibility constraints, and (4) non-identically distributed types of agents. Our work addresses the first three of these four challenges.

We study the problem of scheduling jobs on machines where the runtime of a job on a machine is that machine's private information. The prior over runtimes is a product distribution that is symmetric with respect to the machines (but not necessarily symmetric with respect to the jobs). *Ex ante*, i.e., before the job sizes are instantiated, the machines appear identical; *ex post*, i.e., after the job sizes are realized, the machines are distinct and job runtimes are unrelated. The makespan objective is to schedule the jobs on machines so as to minimize the time at which the last machine completes all of its assigned jobs. Our goal is a prior-independent approximation of the second-best solution for the makespan objective.

To gain intuition for the makespan objective, consider why the simple and incentive compatible VCG mechanism fails to produce a good solution in expectation. The VCG mechanism for scheduling minimizes the total work done by all of the machines and accordingly places every job on its best machine. Note that because the machines are a priori identical, this is an i.i.d. uniformly random machine for every job. Therefore, in expectation, every machine gets an equal number of jobs. Furthermore, every job simultaneously has its smallest size possible. However, the maximum load in terms of the number of jobs per machine and so also the makespan can be quite large. The distribution of jobs across machines is akin to the distribution of balls into bins in the standard balls-in-bins experiment—when the number of balls and bins is equal, the maximum loaded bin contains $\Theta(\log n / \log \log n)$ balls with high probability even though the average load is 1.

Our designed mechanism must prevent the above balls-in-bins style behavior. Consider a variant of VCG that we call the *bounded overload* mechanism. The bounded overload mechanism minimizes the total work with the additional feasibility constraint that the load (i.e., number of jobs scheduled) of any machine is bounded to be at most a c factor more than the average load. This mechanism is “maximal in range”, i.e., it is simply the VCG mechanism with a restricted space of feasible outcomes; it is therefore incentive compatible. Moreover, the bounded overload mechanism can be viewed as belonging to a class of “supply limiting” mechanisms (cf. the prior-independent supply-limiting approximation mechanism of [26] for multi-item revenue maximization).

While the bounded overload mechanism evens out the number of jobs per machine, an individual job may end up having a running time far larger than that on its best machine. The crux of our analysis is to show that this does not hurt the expected makespan of our schedule relative to an ideal setting where every job assumes its minimum size. Our analysis of job sizes has two components. First we show that every job

with high probability gets assigned to one of its best machines. Second, we show that the running time of a job on its i th best machine can be related within a factor depending on i to its running time on its best machine. These components together imply that the bounded overload mechanism simultaneously obtains a schedule that is balanced in terms of the number of jobs per machine and where every job has a small size (in comparison to the best possible for that job). This is sufficient to imply a constant factor approximation to expected makespan when the number of jobs is proportional to the number of machines.

The second component of our analysis of job sizes in the bounded overload mechanism entails relating different order statistics of (arbitrary) i.i.d. distributions, a property that may have broader applications. In particular, letting $X[k:n]$ denote the k th minimum out of n independent draws from a distribution, we show that for any k and n , $X[k:n]$ is nearly stochastically dominated by an exponential function of k times $X[1:n/2]$. In simple terms, the minimum out of a certain number of draws cannot be arbitrarily smaller than the k th minimum out of twice as many draws.

As an intermediary step in our analysis we bound the performance of our approximation mechanism with respect to the first-best solution with half the machines (recall, machines are a priori identical). Within the literature on prior-independent revenue maximization this approach closely resembles the classical Bulow-Klemperer theorem [4]. For auctioning k units of a single-item to n agents (with values drawn i.i.d. from a “nice” distribution), the revenue from welfare maximization exceeds the optimal revenue from $n - k$ agents. In other words, a simple prior-independent mechanism with extra competition (namely, k extra agents) is better than the prior-optimal mechanism for expected revenue. Our result is similar: when the number of jobs is at most the number of machines and machines are a priori identical, we present a prior-independent mechanism that is a constant approximation to makespan with respect to the first-best (and therefore also with respect to the second-best) solution with half as many machines. Unlike the Bulow-Klemperer theorem we place no assumptions the distribution of jobs on machines besides symmetry with respect to machines.

To design scheduling mechanisms for the case where the number of jobs is large relative to the number of machines we can potentially take advantage of the law of large numbers. If there are many more large jobs (i.e., jobs for which the best of the machines’ runtimes is significant) then assigning jobs to machines to minimize total work will produce a schedule where the maximum work on any machine is concentrated around its expectation; moreover, the expected load of any machine in the schedule that minimizes total work is at most the expected load of any machine in the schedule that minimizes makespan.

On the other hand, if there are a moderate number, e.g., proportional to the number of machines, of jobs with very large runtimes on all machines, both the minimum work mechanism and the bounded overload mechanism can fail to have good expected makespan. For the bounded overload mechanism, although the distribution of jobs across machines is more-or-less even, the distribution of the few “worst” jobs that contribute the most to the makespan may be highly uneven. Indeed, for a distribution where the expected number of large jobs is about the same as the number of machines, the bounded overload mechanism exhibits the same bad balls-in-bins behavior as the minimum work mechanism.

The problem above is that the existence of many small, but relatively easy to schedule jobs, prevents the bounded overload mechanism from working. To solve this problem we employ a two stage approach. The first stage acts as a sieve and schedules the small jobs to minimize total work and while leaving the large jobs unscheduled. Then in the second stage the bounded overload mechanism is run on the unscheduled jobs. With the proper parameter tunings (i.e., job size threshold for the sieve and partitioning of machines to the two stages) this mechanism gives a schedule with approximately optimal expected makespan. We give two parameter tunings and analyses, one which gives an $O(\sqrt{\log m})$ approximation and the other that gives an $O((\log \log m)^2)$ approximation under a certain tail condition on the distribution of job sizes (satisfied, for example, by all monotone hazard rate distributions).

The proper tuning of the parameters of the mechanism require knowledge of a single order statistic of the size distribution, namely the expected size of a job on its best out of k machines for an appropriate value of k , to decide which jobs get scheduled in which stage. This statistic can be easily estimated as the mechanism is running by using the reports of a small fraction of the machines as a “market analysis.” To keep our exposition and analysis simple, we skip this detail and assume that the statistic is known.

Related work

There is a large body of work on prior-free mechanism design for the makespan objective. This work does not assume a prior distribution, instead it looks at worst-case approximation of the first-best solution (i.e., the optimal makespan without incentive constraints). The problem was introduced by Nisan and Ronen [25] who showed that the minimum work (a.k.a. VCG) mechanism gives an m -approximation to makespan (where m is the number of machines). They gave a lower bound of two on the worst case approximation factor of any dominant strategy mechanism for unrelated machine scheduling. They conjectured that the best worst-case approximation is indeed $\Theta(m)$. Following this work, a series of papers presented better lower bounds for deterministic as well as randomized mechanisms [8, 7, 18, 24]. Ashlagi, Dobzinski and Lavi [3] recently proved a restricted version of the Nisan-Ronen conjecture by showing that no *anonymous* deterministic dominant-strategy incentive-compatible mechanism can achieve a factor better than m . This lower bound suggests that the makespan objective is fundamentally incompatible with incentives in prior-free settings. In this context, our work can be viewed as giving a meaningful approach for obtaining positive results that are close to prior-free for a problem for which most results are very negative.

Given these strong negative results, several special cases of the problem have been studied. Lavi and Swamy [19] give constant factor approximations when job sizes can take on only two different values. Lu and Yu [22, 21, 20] consider the problem over two machines, and give approximation ratios strictly better than 2.

Related machine scheduling is the special case where the runtime of a job on a machine is the product of the machine’s private speed and the job’s public length. Importantly, the private information of each machine in a related machine scheduling problem is single-dimensional, and the total length of the jobs assigned to any given machine in the makespan minimizing schedule is monotone in the machine’s speed. This monotonicity implies that the related machine makespan objective is incentive compatible (i.e., the price of anarchy is one). For this reason work on related machine scheduling has focused on computational tractability. Archer and Tardos [2] give a constant approximation mechanism and Dhangwotnotai et al. [12] give an incentive compatible polynomial time approximation scheme thereby matching the best approximation result absent incentives. There are no known approximation-preserving black-box reductions from mechanism design to algorithm design for related machine scheduling; moreover, in the Bayesian model Chawla, Immorlica, and Lucier [6] recently showed that the makespan objective does not admit black-box reductions of the form that Hartline and Lucier [16] showed exist for the objective of social welfare maximization.

Another line of work studies the makespan objective subject to an envy-freeness constraint instead of the incentive-compatibility constraint. A schedule and payments (to the machines) are envy free if every machine prefers its own assignment and payment to that of any other machine. Mu’alem [23] introduced the envy-free scheduling problem for makespan. Cohen et al. [10] gave a polynomial time algorithm for computing an envy-free schedule that is an $O(\log m)$ approximation to the first-best makespan (i.e., the optimal makespan absent envy-freeness constraints). Fiat and Levavi [14] complement this by showing that the optimal envy-free makespan (a.k.a. second-best makespan) can be an $\Omega(\log m)$ factor larger than the first-best makespan.

2 Preliminaries and main results

We consider the scheduling of n jobs on m unrelated machines where the running time of a job on a machine is drawn from a distribution. A *schedule* is an assignment of each job to exactly one machine. The *load* of a machine is the number of jobs assigned to it. The *load factor* is the average number of jobs per machine and is denoted $\eta = n/m$. The *work* of a machine is the sum of the runtimes of jobs assigned to it. The *total work* is the sum of the works of each machine. The *makespan* is the most work assigned to any machine.

The vector of running times for each of the jobs on a given machine is that machine's private information. A scheduling mechanism may solicit this information from the machines, may make payments to the machines, and must select a schedule of jobs on the machines. A scheduling mechanism is evaluated in the equilibrium of strategic behavior of the machines. A particularly robust equilibrium concept is dominant strategy equilibrium. A scheduling mechanism is *incentive compatible* if it is a dominant strategy for each machine to report its true processing time for each job.

We consider the following simple mechanisms:

minimum work The minimum work mechanism solicits the running times, selects the schedule to minimize the total work and pays each machine its externality, i.e., the difference between the minimum total work when the machine does nothing and the total work of all other machines in the selected schedule.

bounded overload The bounded overload mechanism is parameterized by an overload factor $c > 1$ and is identical to the minimum work mechanism except it optimizes subject to placing at most $c\eta$ jobs on any machine.

sieve / anonymous reserve The sieve mechanism, also known as the anonymous reserve mechanism, is parameterized by a reserve $\beta \geq 0$ and is identical to the minimum work mechanism except that there is a dummy machine added with runtime β for all jobs. Jobs assigned to the dummy machine are considered unscheduled.

sieve and bounded overload The sieve and bounded overload mechanism is parameterized by overload c , reserve β , and a partition parameter δ . It partitions the machines into two sets of sizes $(1 - \delta)m$ and δm . It runs the sieve with reserve β on the first set of machines and runs the bounded overload mechanism with overload c on the unscheduled jobs and the second set of machines.

The above mechanisms are incentive compatible. The minimum work mechanism is incentive compatible as it is a special case of the well known Vickrey-Clarke-Groves (VCG) mechanism which is incentive compatible. The bounded overload mechanism is what is known as a “maximal in range” mechanism and is also incentive compatible (by the VCG argument). The sieve / anonymous reserve mechanism is incentive compatible because the incentives of the agents in the minimum work mechanism are unaffected by the addition of a dummy agent. Finally, the sieve and bounded overload mechanism is incentive compatible because from each machine's perspective it is either participating in the sieve mechanism or the bounded overload mechanism.

The runtimes of jobs on machines are drawn from a product distribution (a.k.a., the *prior*) that is symmetric with respect to the machines. (Therefore, the running times of a job on each machine are i.i.d. random variables.) The distribution of job j on any machine is denoted F_j ; a draw from this distribution is denoted T_j . The *best runtime* of a job is its minimum runtime over all machines, this first order statistic of m random draws from F_j is denoted by $T_j[1:m]$.

Our goal is to exhibit a mechanism that is prior-independent and a good approximation to the expected makespan of the best incentive compatible mechanism for the prior, i.e., the second-best solution. Because both the second-best and the first-best expected makespans are difficult to analyze, we will give our approximation via one of the following two lower bounds on the first-best solution.

expected worst best runtime The expected worst best runtime is the expected value of the best runtime of the job with the longest best runtime, i.e., $\mathbf{E}[\max_j T_j[1:m]]$

expected average best runtime The expected average best runtime is the expected value of the sum of the best runtimes of each job averaged over all machines, i.e., $\mathbf{E}[\sum_j T_j[1:m]]/m$.

Intuitively, the former gives a good bound when the load factor is small, the latter when the load factor is large. We will refer to any of these bounds on the first-best makespan as OPT, with the assumption that which of the bounds is meant, if it is important, is clear from the context.

As an intermediary in our analysis of the makespan of our scheduling mechanisms with respect to OPT, we will give bicriteria results that compare our mechanism's makespan to the makespan of an optimal schedule with fewer machines. This restriction is well defined because the machines are a priori identical. For a given parameter δ , OPT_δ will denote the optimal schedule with δm machines (via bounds as described above). Much of our analysis will be with respect to $\text{OPT}_{1/2}$, i.e., the optimal schedule with half the number of machines.

While it is possible to construct distributions where OPT is much smaller than $\text{OPT}_{1/2}$, for many common distributions they are quite close. In fact, for the class of distributions that satisfy the monotone hazard rate (MHR) condition,¹ OPT and $\text{OPT}_{1/2}$ are always within a factor of four; more generally OPT and OPT_δ are within a factor of $1/\delta^2$ for these distributions. (See proof in Section 5.)

Lemma 2.1 *When the distributions of job sizes have monotone hazard rates the expected worst best and average best runtimes on δm machines are no more than $1/\delta^2$ times the expected worst best and average best runtimes, respectively, on m machines.*

2.1 Main Results

Our main theorems are as follows. When the number of jobs is comparable to the number of machines, i.e., the load factor η is constant, then the bounded overload mechanism is a good approximation to the optimal makespan on $m/2$ machines.

Theorem 2.2 *For n jobs, m machines, load factor $\eta = n/m$, and runtimes distributed according to a machine-symmetric product distribution, the expected makespan of the bounded overload mechanism with overload $c = 7$ is a 200η approximation to the expected worst best runtime, and hence also to the optimal makespan, on $m/2$ machines.*

Corollary 2.3 *Under the assumptions of Theorem 2.2 where additionally the distributions of job sizes have monotone hazard rates, the expected makespan of the bounded overload mechanism with $c = 7$ is a 800η approximation to the expected optimal makespan.*

¹The hazard rate of a distribution F is given by $h(x) = \frac{f(x)}{1-F(x)}$, where f is the probability density function for F ; a distribution F satisfies the MHR condition if $h(x)$ is non-decreasing in x . Many natural distributions such as the uniform, Gaussian, and exponential distributions, satisfy the monotone hazard rate condition. Intuitively, these are distributions with tails no heavier than the exponential distribution.

When the load factor η is large and the job runtimes are identically distributed, the sieve and bounded overload mechanism is a good approximation to the optimal makespan. The following theorems and corollaries demonstrate the sieve and bounded overload mechanism under two relevant parameter settings.

Theorem 2.4 *For n jobs, m machines, and runtimes from an i.i.d. distribution, the expected makespan of the sieve and bounded overload mechanism with overload $c = 7$, partition parameter $\delta = 2/3$, and reserve $\beta = \frac{n}{m \log m} \mathbf{E}[T[1:\frac{\delta}{2}m]]$ is an $O(\sqrt{\log m})$ approximation to the larger of the expected worst best and average best runtime, and hence also to the optimal makespan, on $m/3$ machines. Here T denotes a draw from the distribution on job sizes.*

Corollary 2.5 *Under the assumptions of Theorem 2.4 where additionally the distribution of job sizes has monotone hazard rate, the expected makespan of the sieve and bounded overload mechanism is an $O(\sqrt{\log m})$ approximation to the expected optimal makespan.*

Theorem 2.6 *For $n \geq m \log m$ jobs, m machines, and runtimes from an i.i.d. distribution, the expected makespan of the sieve and bounded overload mechanism with overload $c = 7$, partition parameter $\delta = 1/\log \log m$, and reserve $\beta = \frac{2n}{m \log m} \mathbf{E}[T[1:\frac{\delta}{2}m]]$, is a constant approximation to the larger of the expected worst best and average best runtime, and hence also to the optimal makespan, on $\delta m/2$ machines. Here T denotes a draw from the distribution on job sizes.*

Corollary 2.7 *Under the assumptions of Theorem 2.6 where additionally the distribution of job sizes has monotone hazard rate the expected makespan of the sieve and bounded overload mechanism is a $O((\log \log m)^2)$ approximation to the expected optimal makespan.*

We prove Theorem 2.2 in Section 3 and Theorems 2.4 and 2.6 in Section 4.

2.2 Probabilistic Analysis

Our goal is to show that the simple processes described by the bounded overload and sieve mechanisms result in good makespan and our upper bound on makespan is given by the first order statistics of each job's runtime across the machines. The sieve's performance analysis is additionally governed by the law of large numbers. We describe here basic facts about order statistics and concentration bounds. Additionally we give a number of new bounds, proofs of which are in Section 5.

For random variable X and integer k , we consider the following basic constructions of k independent draws of the random variable. The i th order statistic, or the i th minimum of k draws, is denoted $X[i:k]$. The first order statistic, i.e., the minimum of the k draws, is denoted $X[1:k]$. The k th order statistic, i.e., the maximum of k draws, is denoted $X[k:k]$. Finally, the sum of k draws is denoted $X[\Sigma k]$. We include the possibility that i or k can be random variables. We also allow the notation to cascade, e.g., for the special case where the jobs are i.i.d. from F the lower bounds on OPT are $T[1:m][n:n]$ and $T[1:m][\Sigma n]/m$ for the expected worst best and average best runtime, respectively, and T drawn from F .

We will use the following forms of Chernoff-Hoeffding bounds in this paper. Let $X = \sum_i X_i$, where $X_i \in [0, B]$ are independent random variables. Then, for all $\epsilon \geq 1$,

$$\Pr[X > (1 + \epsilon) \mathbf{E}[X]] < \exp\left(\frac{-\epsilon \mathbf{E}[X]}{3B}\right) < \exp\left(\frac{-(1+\epsilon) \mathbf{E}[X]}{6B}\right)$$

Our analysis often involves relating different order statistics of a random variable (e.g. how does the size of a job on its best machine compare to that on its second best machine). We relate these different order

statistics via the stochastic dominance relation. This is useful in our analysis because stochastic dominance is preserved by the max and sum operators. We say that a random variable X is stochastically dominated by another random variable Y if for all t , $\Pr[X \leq t] \geq \Pr[Y \leq t]$. Stochastic dominance is equivalent to being able to couple the two random variables X and Y so that X is always smaller than Y .

Below, the first lemma relates the i th order statistic over some number of draws to the first order statistic over half the draws. The second relates the minimum over several draws of a random variable to a single draw of that variable. The third relates the maximum over multiple draws of a random variable to an appropriate sum over those draws. These lemmas are proved in Section 5.

Lemma 2.8 *Let X be any nonnegative random variable and m and $i \leq m$ be arbitrary integers. Let α be defined such that $\Pr[X \leq \alpha] = 1/m$ (or for discontinuous distributions, $\alpha = \sup\{z : \Pr[X \leq z] < 1/m\}$). Then $X[i:m]$ is stochastically dominated by $\max(\alpha, X[1:m/2][4^i:4^i])$.*

Lemma 2.9 *For a random variable X whose distribution satisfies the monotone hazard rate condition, X is stochastically dominated by $rX[1:r]$.*

Lemma 2.10 *Let K_1, \dots, K_n be independent and identically distributed integer random variables such that for some constant $c > 1$, we have $K_j \geq c$, and let W_1, \dots, W_n be arbitrary independent nonnegative variables. Then,*

$$\mathbf{E}[\max_j W_j[K_j:K_j]] \leq \frac{c}{c-1} \mathbf{E}[K_1] \mathbf{E}[\max_j W_j].$$

We will analyze the expected makespan of a mechanism as the maximum over a number of correlated real-valued random variables. The correlation among these variables makes it difficult to understand and bound the makespan. Our approach will be to replace these random variables with an ensemble of independent random variables that have the same marginal distributions. Fortunately, this operation does not change the expected maximum by too much. Our next lemma relates the expected maximum over an arbitrary set of random variables to the expected maximum over a set of independent variables with the same marginal distributions. It is a simple extension of the correlation gap results of Aggarwal et al. [1], Yan[28], and Chawla et al. [5].

Lemma 2.11 *Let X_1, \dots, X_n be arbitrary correlated real-valued random variables. Let Y_1, \dots, Y_n be independent random variables defined so that the distribution of Y_i is identical to that of X_i for all i . Then, $\mathbf{E}[\max_j X_j] \leq \frac{e}{e-1} \mathbf{E}[\max_j Y_j]$.*

3 The bounded overload mechanism

Recall that the bounded overload mechanism minimizes the total work subject to the additional feasibility constraint that every machine is assigned at most $c\eta$ jobs. In this section we prove that the expected makespan of the bounded overload mechanism, with the overload set to $c = 7$, is a 200η factor approximation to the expected best worst runtime and thus to the optimal makespan.

Intuitively the bounded overload mechanism tries to achieve two objectives simultaneously: (1) keep the size of every job on the machine its schedule to be close to its size on its best machine, but also (2) evenly distribute the jobs across all the machines. Recall, that the minimum work mechanism achieves the first objective exactly, but fails on the second objective. Due to the independence between jobs, the number of jobs on each machine may be quite unevenly distributed. In contrast, the bounded overload

mechanism explicitly disallows uneven assignments of jobs and therefore the main issue to address in its analysis is whether it satisfies the first objective, i.e., that the sizes of the jobs are close to what they are in the minimum work mechanism.

To setup for the proof of Theorem 2.2 consider the following definitions that describe the outcome of the bounded overload mechanism and the worst best runtime on $m/2$ machines (which bounds the optimal makespan on $m/2$ machines). Let T_j denote a random variable drawn according to job j 's distribution of runtimes F_j . Let B_j denote the job's best runtime out of $m/2$ machines, i.e., $B_j = T_j[1:m/2]$, the first order statistic of $m/2$ draws. The expected worst best runtime on $m/2$ machines is $\mathbf{E}[\max_j B_j]$. The bounded overload mechanism considers placing each job on one of m machines. These runtimes of job j drawn i.i.d. from F_j impose a (uniformly random) ordering over the machines starting from the machine that is "best" for j to the one that is "worst"; this is j 's preference list. Let $T_j[r:m]$ denote the size of job j on the r th machine in this ordering (also called the job's r th favorite machine). Let R_j be a random variable to denote the rank of the machine that job j is placed on by the bounded overload mechanism. As each machine is constrained to receive at most $c\eta$ jobs, the expected makespan of bounded overload is $c\eta \mathbf{E}[\max_j T_j[R_j:m]]$. We will bound this quantity in terms of $\mathbf{E}[\max_j B_j]$.

There are three main parts to our argument. First, we note that the R_j s are correlated across different j 's, and so are the $T_j[R_j:m]$ s. This makes it challenging to directly analyze $\mathbf{E}[\max_j T_j[R_j:m]]$. We use Lemma 2.11 to replace the R_j s in this expression by independent random variables with the same marginal distributions. We then show that the marginal distributions can be bounded by simple geometric random variables \tilde{R}_j . To do so, we introduce another procedure for assigning jobs to machines that we call the last entry procedure. The assignment of each job under the last entry procedure is no better than its assignment under bounded overload. On the other hand, the ranks of the machines to which jobs are allocated in the last entry procedure are geometric random variables with a bounded failure rate. Finally, we relate the runtimes $T_j[\tilde{R}_j:m]$ to the optimal runtimes B_j using Lemma 2.8.

We begin by describing the last entry procedure.

last entry In order to schedule job j , we first apply the bounded overload mechanism BO_c to all jobs other than j . We then place j on the first machine in its preference list that has fewer than $c\eta$ jobs. Let L_j denote the rank of the machine to which j gets allocated.

We now make a few observations about the ranks L_j realized by the last entry procedure.

Lemma 3.1 *The runtime of any job j in bounded overload is no worse than its runtime in the last entry procedure. That is, $R_j \leq L_j$.*

Proof: Fix any instantiation of jobs' runtimes over machines. Consider the assignment of job j in the last entry procedure, and let $\text{LE}(j)$ denote the schedule where all of the jobs but j are scheduled according to bounded overload and j is scheduled according to the last entry procedure. Since the bounded overload mechanism minimizes total work, the total runtime of all of the jobs in BO_c is no more than the total runtime of all of the jobs in $\text{LE}(j)$. On the other hand, the total runtime of all jobs except j in $\text{LE}(j)$ is no more than the total runtime of all jobs except j in BO_c . This immediately implies that j 's runtime in bounded overload is no more than its runtime in last entry. Since this holds for any fixed instantiation of runtimes, we have $R_j \leq L_j$. ■

Next, we show that the rank L_j of a job j in last entry is stochastically dominated by a geometric random variable \tilde{R}_j that is capped at $\lceil \frac{m}{c} \rceil$. Note that L_j is at most $\lceil \frac{m}{c} \rceil$ since $\lceil \frac{m}{c} \rceil$ machines can accommodate $\lceil \frac{m}{c} \rceil c\eta \geq n$ jobs and therefore last entry will never have to send a job to anything worse than its $\lceil \frac{m}{c} \rceil$ th

favorite machine. The random variable \tilde{R}_j also lives in $\{1, \dots, \lceil \frac{m}{c} \rceil\}$, and is drawn independently for all j as follows: for $i \in \{1, \dots, \lceil \frac{m}{c} \rceil - 1\}$, we have $\Pr[\tilde{R}_j = i] = \frac{1-1/c}{c^{i-1}}$; and the remaining probability mass is on $\lceil \frac{m}{c} \rceil$.

Lemma 3.2 *The rank L_j of a job j in last entry is stochastically dominated by \tilde{R}_j , and so the runtime of job j in last entry is stochastically dominated by $T_j[\tilde{R}_j:m]$.*

Proof: We use the principle of deferred decisions. In order to schedule j , the last entry procedure first runs bounded overload on all of the jobs other than j . This produces a schedule in which at most a $\frac{1}{c}$ fraction of the machines have all of their slots occupied. Conditioned on this schedule, job j 's preference list over machines is a uniformly random permutation. So the probability (over the draw of j 's runtimes) that job j 's favorite machine is fully occupied is at most $1/c$. Likewise, the probability that the job's two most favorite machines are both occupied is at most $1/c^2$, and so on. Therefore, the rank of the machine on which j is eventually scheduled is dominated by a geometric random variable with failure rate $1/c$. ■ Lemmas 3.1

and 3.2 yield the following corollary.

Corollary 3.3 *For all j , the runtime $T_j[R_j:m]$ of job j in bounded overload is stochastically dominated by $T_j[\tilde{R}_j:m]$.*

The benefit of relating $T_j[R_j:m]$ s with $T_j[\tilde{R}_j:m]$ s is that while the former are correlated random variables, the latter are independent, because the \tilde{R}_j 's are picked independently. Corollary 3.3 implies that we can replace the former with the latter, gaining independence, while losing only a constant factor in expected makespan.

Corollary 3.4 $\mathbf{E}[\max_j T_j[R_j:m]]$ is no more than $e/(e-1)$ times $\mathbf{E}[\max_j T_j[\tilde{R}_j:m]]$.

The final part of our analysis relates the $T_j[\tilde{R}_j:m]$ s to the B_j s. A natural inequality to aim for is to bound $\mathbf{E}[T_j[\tilde{R}_j:m]]$ from above by a constant times $\mathbf{E}[B_j]$ for each j . Unfortunately, this is not enough for our purposes: note that our goal is to upper bound $\mathbf{E}[\max_j T_j[\tilde{R}_j:m]]$ in terms of $\mathbf{E}[\max_j B_j]$. Thus we proceed to show that $T_j[\tilde{R}_j:m]$ is stochastically dominated by a maximum among some number of copies of B_j . We apply Lemma 2.8 (stated in Section 2 and proved in Section 5) to the random variable $T_j[i:m]$ for this purpose. Define $\alpha_j = \sup\{t : F_j(t) < 1/m\}$. Then the lemma shows that $T_j[i:m]$ is stochastically dominated by $\max(\alpha_j, B_j[4^i:4^i])$.

Let D_j be defined as $4^{\tilde{R}_j}$. Note that $\mathbf{E}[D_j]$ can be bounded by a constant whenever $c > 4$ (this upper bound is obtained by treating \tilde{R}_j as a geometric random variable without being capped at $\lceil \frac{m}{c} \rceil$). Then Lemma 2.8 implies the following corollary.

Lemma 3.5 $T_j[\tilde{R}_j:m]$ is stochastically dominated by $\max(\alpha_j, B_j[D_j:D_j])$.

We are now ready to prove the main theorem of this section.

Theorem 2.2 *For n jobs, m machines, load factor $\eta = n/m$, and runtimes distributed according to a machine-symmetric product distribution, the expected makespan of the bounded overload mechanism with overload $c = 7$ is a 200η approximation to the expected worst best runtime, and hence also to the optimal makespan, on $m/2$ machines.*

Proof: The proof follows from the following series of inequalities that we explain below. First we have $\text{Makespan}(\text{BO}_c) \leq c\eta \mathbf{E}[\max_j T_j[R_j:m]]$ by the fact that BO_c schedules at most $c\eta$ jobs per machine

$$\begin{aligned}
\frac{e-1}{e} \mathbf{E} \left[\max_j T_j[R_j:m] \right] &\leq \mathbf{E} \left[\max_j T_j[\tilde{R}_j:m] \right] \\
&\leq \mathbf{E} \left[\max_j (\max(\alpha_j, B_j[D_j:D_j])) \right] \\
&\leq \mathbf{E} \left[\max_j (\alpha_j + B_j[D_j:D_j]) \right] \\
&\leq \max_j \alpha_j + \mathbf{E} \left[\max_j B_j[D_j:D_j] \right] \\
&\leq 2\text{OPT}_{1/2} + \frac{4}{4-2} \mathbf{E}[D_j] \mathbf{E}[\max_j B_j] \\
&\leq \left(2 + 8 \frac{c-1}{c-4} \right) \text{OPT}_{1/2}.
\end{aligned}$$

The first of the inequalities follows from Lemma 3.4, the second from Lemma 3.5, the third from noting that the maximum of non-negative random variables is upper bounded by their sum, and the last by the definition of $\text{OPT}_{1/2}$, along with the fact that $\mathbf{E}[D_j] \leq 4 \frac{c-1}{c-4}$. For the fifth inequality we use Lemma 2.10 to bound the second term. For the first term in that inequality consider the job j that has the largest α_j . For this job, the probability that its size on all of the $m/2$ machines in $\text{OPT}_{1/2}$ is at least α_j is $(1 - F_j(\alpha_j))^{m/2} \geq (1 - 1/m)^{m/2} \geq 1/2$ by the definition of α_j . So $\text{OPT}_{1/2} \geq \max_j \alpha_j/2$.

The final approximation factor therefore is $c\eta \frac{e}{e-1} \left(2 + 8 \frac{c-1}{c-4} \right)$ for all $c > 4$. At $c = 7$, this evaluates to a factor 200η approximation. ■

4 The sieve and bounded overload mechanism

We will now analyze the performance of the sieve and bounded overload mechanisms under the assumption that the jobs are a priori identical. Let us consider the sieve mechanism first. Recall that this is essentially the minimum work mechanism where every job is assigned to its best machine, except that jobs with a size larger than β on every machine are left unscheduled. The bound of β on the size of scheduled jobs allows us to employ concentration results to bound the expected makespan of the mechanism. Changing the value of β allows us to tradeoff the makespan of the mechanism with the number of unscheduled jobs.

Lemma 4.1 *For $k < \log m$, the expected makespan of the sieve mechanism with $\beta = \frac{n \mathbf{E}[T[1:m]]}{km}$ is no more than $O(\log m/k)$ times the expected average best runtime, and hence also the expected optimal makespan. The expected number of jobs left unscheduled by the mechanism is km .*

Proof: Let us first consider the expected total work of any single machine, that is the expected total size of jobs scheduled on that machine. Let Y_{ij} be a random variable that takes on the value 0 if job j is not scheduled on machine i , and takes on the size of j on machine i if the job is scheduled on that machine. The probability that j is scheduled on i is no more than $1/m$; its expected size on i conditioned on being scheduled is at most $\tau = \mathbf{E}[T[1:m]]$. Therefore, $\mathbf{E}[\sum_j Y_{ij}] \leq \frac{n\tau}{m}$, which in turn is at most the average best runtime.

Note that the Y_{ij} 's are independent and bounded random variables. So we can apply Chernoff-Hoeffding bounds and use $\beta = \frac{n\tau}{km}$ to get

$$\Pr \left[\sum_j Y_{ij} > \frac{7 \log m}{k} \text{OPT} \right] \leq \Pr \left[\sum_j Y_{ij} > \frac{7 \log m}{k} \mathbf{E} \left[\sum_j Y_{ij} \right] \right] < \exp \left(-\frac{1}{3} \frac{6 \log m}{k} \frac{n\tau}{\beta m} \right) = \frac{1}{m^2}.$$

Taking the union bound over the m machines, we get that with probability $1 - 1/m$, the makespan of the sieve mechanism is at most $O(\log m/k)$ times OPT.

We will now convert this tail probability into a bound on the expected makespan. Let γ denote the factor by which the expected makespan of the mechanism exceeds OPT. Remove all jobs with best runtimes greater than β from consideration and consider creating sieve's schedule by assigning each of the leftover jobs to their best machine (minimizing total work) one-by-one in decreasing order of best runtime, until the makespan exceeds $\frac{7}{k} \log m$ times OPT. This event happens with a probability at most $1/m$. When this event happens, we are left with a smaller set of jobs; conditioned on being left over at this point, these jobs have a smaller best runtime than the average over all scheduled jobs. Thus the expected makespan for scheduling them will be at most γOPT . So we get $\gamma \leq 7 \log m/k + \gamma/m$, i.e., $\gamma = O(\log m/k)$. This implies the first part of the lemma.

We now prove the second part of the lemma, i.e., the expected number of jobs left unscheduled is km . Note that β exceeds a job's expected best runtime by a factor of n/km . Thus by applying Markov's inequality, we get the probability of a job's best runtime being larger than β to be at most km/n . Hence the expected number of jobs with best runtime larger than β is km . ■

Next we will combine the sieve mechanism with the bounded overload mechanism. We consider two different choices of parameters. Note that if in expectation the sieve mechanism leaves km jobs unscheduled, using the bounded overload mechanism to schedule these jobs over a set of $\Omega(m)$ machines gives us an expected makespan that is at most $O(k)$ larger than the expected optimal makespan on that number of machines. In order to balance this with the makespan achieved by sieve, we pick $k = \sqrt{\log m}$. This gives us Theorem 2.4.

Theorem 2.4 *For n jobs, m machines, and runtimes from an i.i.d. distribution, the expected makespan of the sieve and bounded overload mechanism with overload $c = 7$, partition parameter $\delta = 2/3$, and reserve $\beta = \frac{n}{m \log m} \mathbf{E}[T[1:\frac{\delta}{2}m]]$, is an $O(\sqrt{\log m})$ approximation to the larger of the worst best runtime and the average best runtime, and hence also to the optimal makespan, on $m/3$ machines. Here T denotes a draw from the distribution on job sizes.*

Proof: For the choice of parameters in the theorem statement, we use $m/3$ of the m machines for the sieve mechanism, and the remainder for the bounded overload mechanism. The expected makespan of the overall mechanism is no more than the sum of the expected makespans of the two constituent mechanisms. Lemma 4.1 implies that the expected makespan of the sieve mechanism is $O(\sqrt{\log m})$ times $\text{OPT}_{1/3}$, and the load factor for the bounded overload mechanism is also $O(\sqrt{\log m})$. Theorem 2.2 then implies that the expected makespan of the bounded overload mechanism is also $O(\sqrt{\log m})$ times $\text{OPT}_{1/3}$. ■

If we partition the machines across the sieve and the bounded overload mechanisms roughly equally, then Theorem 2.4 gives us the optimal choice for the parameter β . A different possibility is to perform a more aggressive screening of jobs by using a smaller β , while comparing our performance against a more heavily penalized optimal mechanism – one that is allowed to use only a δ fraction of the machines.

Theorem 2.6 For $n \geq m \log m$ jobs, m machines, and runtimes from an i.i.d. distribution, the expected makespan of the sieve and bounded overload mechanism with overload $c = 7$, partition parameter $\delta = 1/\log \log m$, and reserve $\beta = \frac{2n}{m \log m} \mathbf{E}[T[1:\frac{\delta}{2}m]]$, is a constant approximation to the larger of the worst best runtime and the average best runtime, and hence also to the optimal makespan, on $\delta m/2$ machines. Here T denotes a draw from the distribution on job sizes.

Proof: We will show that the expected makespan of the sieve mechanism is at most a constant times the average best runtime on $\delta m/2$ machines, and the expected number of unscheduled jobs is $O(\delta m)$. The current theorem then follows by applying Theorem 2.2.

Let us analyze the expected makespan of the sieve mechanism first. Let $\tau = \mathbf{E}[T[1:\frac{\delta}{2}m]]$. Then we can bound $\text{OPT}_{\delta/2}$ as $\text{OPT}_{\delta/2} \geq \frac{2n\tau}{\delta m}$. As in the proof of Lemma 4.1, let Y_{ij} be a random variable that takes on the value 0 if job j is not scheduled on machine i , and takes on the size of j on machine i if the job is scheduled on that machine. Then,

$$\mathbf{E} \left[\sum_j Y_{ij} \right] \leq \frac{n}{(1-\delta)m} \mathbf{E}[T[1:(1-\delta)m]] \leq \frac{n\tau}{(1-\delta)m} \leq \frac{\delta}{2(1-\delta)} \text{OPT}_{\delta/2}.$$

Applying Chernoff-Hoeffding bounds we get

$$\Pr \left[\sum_j Y_{ij} > 2\text{OPT}_{\delta/2} \right] \leq \Pr \left[\sum_j Y_{ij} > 4(1/\delta - 1) \mathbf{E} \left[\sum_j Y_{ij} \right] \right] < \exp \left(-\frac{1}{\delta} \frac{n\tau}{(1-\delta)m} \frac{1}{\beta} \right) \leq m^{-1/2\delta}.$$

Here we used $\beta = 2n\tau/m \log m$. Taking the union bound over the m machines, we get that with probability $o(1)$, the makespan of the sieve mechanism is at most twice $\text{OPT}_{\delta/2}$. Once again, as in the proof of Lemma 4.1 we can convert this tail bound into a constant factor bound on the expected makespan.

Now let us consider the jobs left unscheduled. For any given job, we will compute the probability that its runtime on all of the $(1-\delta)m$ machines is larger than β . Because β is defined in terms of $T[1:\frac{\delta}{2}m]$, we will consider the machines in batches of size $\delta m/2$ at a time. Using Markov's inequality, the probability that the job's runtime exceeds β on all machines in a single batch is at most $m \log m/2n$. There are $2(1/\delta - 1)$ batches in all, so the probability that a job remains unscheduled is at most $(m \log m/n)(2^{2(1-1/\delta)})$, which by our choice of δ is $O(\delta m/n)$. ■

5 Deferred proofs

In this section we prove the bounds for random variables and order statistics from Section 2.2.

Lemma 2.8 Let X be any nonnegative random variable, and $m, i \leq m$ be arbitrary integers. Let α be defined such that $\Pr[X \leq \alpha] = 1/m$ (or for discontinuous distributions, $\alpha = \sup\{z : \Pr[X \leq z] < 1/m\}$). Then $X[i:m]$ is stochastically dominated by $\max(\alpha, X[1:m/2][4^i:4^i])$.

Proof: Let F be the cumulative distribution function of X . We prove this by showing that $X[i:m]$ is “almost” stochastically dominated by $X[1:m/2][4^i:4^i]$; specifically, we show that for all $t \geq \alpha$,

$$\Pr[X[i:m] > t] \leq \Pr[X[1:m/2][4^i:4^i] > t].$$

To prove this inequality, we will define a process for instantiating the variables $X[i:m]$ and $X[1:m/2][4^i:4^i]$ in a correlated fashion such that the former is always larger than the other.

$X[1:m/2][4^i:4^i]$ is a statistic based on $4^i m/2$ independent draws of the random variable X . Consider partitioning these draws into $4^i/2$ groups of size m each. We then randomly split each group into two smaller groups, which we will refer to as blocks, of size $m/2$ each. Define a good event \mathcal{G} to be the event that at least one of these $4^i/2$ groups get split such that the i smallest runtimes in it all fall into the same block. If event \mathcal{G} occurs, arbitrarily choose one group which caused event \mathcal{G} , and for all k define $X[k:m]$ to be the k th min from this group. Otherwise, select an arbitrary group to define the $X[k:m]$. Note that since we split the groups into blocks randomly, and this is independent of the drawn runtimes in the groups, $X[k:m]$ has the correct distribution, both when \mathcal{G} occurs and does not occur. Define the minimum from each of the 4^i blocks to be a draw of $X[1:m/2]$. Thus, whenever \mathcal{G} occurs, the probability that the $X[1:m/2][4^i:4^i] > t$ is at least the probability that $X[i+1:m] > t$. We have that

$$\begin{aligned} \Pr[X[1:m/2][4^i:4^i] > t] &\geq \Pr[\mathcal{G}] \cdot \Pr[X[i+1:m] > t] \\ &= \left(\Pr[\mathcal{G}] \cdot \frac{\Pr[X[i+1:m] > t]}{\Pr[X[i:m] > t]} \right) \cdot \Pr[X[i:m] > t]. \end{aligned}$$

We now show that $\left(\Pr[\mathcal{G}] \cdot \frac{\Pr[X[i+1:m] > t]}{\Pr[X[i:m] > t]} \right) \geq 1$ whenever $F(t) \geq 1/m$, which completes our proof of the lemma. Note that

$$\begin{aligned} \frac{\Pr[X[i+1:m] > t]}{\Pr[X[i:m] > t]} &= \frac{\sum_{k=0}^i \binom{m}{k} F(t)^k (1-F(t))^{m-k}}{\sum_{k=0}^{i-1} \binom{m}{k} F(t)^k (1-F(t))^{m-k}} \\ &= 1 + \frac{\binom{m}{i} F(t)^i (1-F(t))^{m-i}}{\sum_{k=0}^{i-1} \binom{m}{k} F(t)^k (1-F(t))^{m-k}}, \end{aligned}$$

which we can see is an increasing function of $F(t)$. Thus in the range $F(t) \geq 1/m$, it attains its minimum precisely at $F(t) = 1/m$. Substituting $F(t) = 1/m$ into the above, and using standard approximations for $\binom{m}{k}$ (namely $\left(\frac{m}{k}\right)^k \leq \binom{m}{k} \leq \left(\frac{me}{k}\right)^k$), we have

$$\begin{aligned} \frac{\Pr[X[i+1:m] > t]}{\Pr[X[i:m] > t]} &\geq 1 + \frac{\left(\frac{m}{i}\right)^i \left(\frac{1}{m}\right)^i \left(1 - \frac{1}{m}\right)^{m-i}}{\left(1 - \frac{1}{m}\right)^m + \sum_{k=1}^{i-1} \left(\frac{me}{k}\right)^k \left(\frac{1}{m}\right)^k \left(1 - \frac{1}{m}\right)^{m-k}} \\ &\geq 1 + \frac{i^{-i}}{1 + (i-1) \cdot \max_k \left(\frac{e}{k}\right)^k} \geq 1 + \frac{i^{-i}}{1 + (i-1)e}. \end{aligned}$$

It suffices to show that this last quantity, when multiplied with $\Pr[\mathcal{G}]$, is at least 1. We consider the complement of event \mathcal{G} , call it even \mathcal{B} . The event \mathcal{B} occurs only when none of the $4^i/2$ groups split favorably. The probability that a group splits favorably (for $i \geq 1$) is $2 \cdot \binom{m-i}{m/2-i} / \binom{m}{m/2} \geq 2^{-(i-1)}$. So we can see that $\Pr[\mathcal{B}] \leq (1 - 2^{-(i-1)})^{4^i/2} \leq e^{-(4/2)^i}$, and thus $\Pr[\mathcal{G}] \geq 1 - e^{-(4/2)^i}$. It can be verified that $(1 - e^{-(4/2)^i}) \cdot \left(1 + \frac{i^{-i}}{1+(i-1)e}\right) \geq 1$. ■

Lemma 2.9 *For a random variable X whose distribution satisfies the monotone hazard rate condition, X is stochastically dominated by $rX[1:r]$.*

Proof: The hazard rate function is related to the cumulative distribution function as $\Pr[X \geq t] = e^{-\int_0^t h(z) dz}$. Likewise, we can write:

$$\Pr[rX[1:r] \geq t] = \Pr[X[1:r] \geq t/r] = (e^{-\int_0^{t/r} h(z) dz})^r = e^{-r \int_0^{t/r} h(z) dz}.$$

In order to prove the lemma, we need only show that $\int_0^t h(z) dz \geq r \cdot \int_0^{t/r} h(z) dz$. Since the hazard rate function $h(z)$ is monotone, the function $\int_0^t h(z) dz$ is a convex function of t . The required inequality follows from the definition of convexity. ■

Lemma 2.10 *Let K_1, \dots, K_n be independent and identically distributed integer random variables such that for some constant $c > 1$, we have $K_j \geq c$ for all j , and let W_1, \dots, W_n be arbitrary independent nonnegative variables. Then,*

$$\mathbf{E}[\max_j W_j[K_j:K_j]] \leq \frac{c}{c-1} \mathbf{E}[K_1] \mathbf{E}[\max_j W_j].$$

Proof: We consider the following process for generating correlated samples for $\max_j W_j$ and $\max_j W_j[K_j:K_j]$. We first independently instantiate K_j for every j ; recall that these are identically distributed variables. Let $k = \sum_j K_j \geq cn$. Then we consider all possible $n!$ permutations of these instantiated values. For each permutation σ , we make the corresponding number of independent draws of the random variable W_j for all j ; call this set of draws X_σ . In all, we get $kn!$ draws from the distributions, that is, $|\cup_\sigma X_\sigma| = kn!$. Exactly $k(n-1)!$ of these draws belong to any particular j ; denote these by Y_j .

Now, the maximum element out of each of the X_σ sets is an independent draw from the same distribution $\max_j W_j[K_j:K_j]$ is drawn from. We get $n!$ independent samples from that distribution. Call this set of samples X .

Next note that each set Y_j contains $k(n-1)!$ independent draws from the distribution corresponding to W_j . We construct a uniformly random n -dimensional matching over the sets Y_j , and from each n -tuple in this matching we pick the maximum. Each such maximum is an independent draw from the distribution corresponding to $\max_j W_j$, and we get $k(n-1)!$ such samples; call this set of samples Y .

Finally, we claim that $\mathbf{E}[\sum_{y \in Y} y] \geq (1 - 1/c) \mathbf{E}[\sum_{x \in X} x]$, with the expectation taken over the randomness in generating the n -dimensional matching across the Y_j s. The lemma follows, since we have $\mathbf{E}[\sum_{x \in X} x] = n! \mathbf{E}[\max_j W_j[K_j:K_j]]$ as well as

$$\mathbf{E}[\sum_{y \in Y} y] = \mathbf{E}_{\{K_j\}}[k(n-1)! \mathbf{E}[\max_j W_j]] = n! \mathbf{E}[K_j] \mathbf{E}[\max_j W_j].$$

To prove the claim, we call an $x \in X$ “good” if the n -tuple in the matching over $\{Y_j\}$ that it belongs to does not contain any other element of X . Then, $\mathbf{E}[\sum_{y \in Y} y] \geq \mathbf{E}[\sum_{x \in X} x \Pr[x \text{ is “good”}]]$.

Let us compute the probability that some x is “good”. Without loss of generality, suppose that $x \in Y_1$. In order for x to be good, it’s n -tuple must not contain any of the other elements of X from the other Y_j ’s. If we define $x_j = |X \cap Y_j|$, then $\Pr[x \text{ is “good”}]$ is at least $\prod_{j \neq 1} (1 - \frac{x_j}{k(n-1)!})$ where $\sum x_j \leq n!$. This product is minimized when we set one of the x_j s to $n!$ and the rest to 0, and takes on a minimum value of $1 - n/k \geq 1 - 1/c$. ■

Lemma 2.11 *Let X_1, \dots, X_n be arbitrary correlated real-valued random variables. Let Y_1, \dots, Y_n be independent random variables defined so that the distribution of Y_i is identical to that of X_i for all i . Then,*

$$\mathbf{E}[\max_j X_j] \leq \frac{e}{e-1} \mathbf{E}[\max_j Y_j].$$

Proof: We use the following result from [1] (also implicit in [5]). Let U be a universe of n elements, f a monotone increasing submodular function over subsets of this universe, and D a distribution over subsets of U . Let \tilde{D} be a product distribution (that is, every element is picked independently to draw a set from this distribution) such that $\Pr_{S \sim D}[i \in S] = \Pr_{S \sim \tilde{D}}[i \in S]$. Then $\mathbf{E}_{S \sim D}[f(S)] \leq \frac{e}{e-1} \mathbf{E}_{S \sim \tilde{D}}[f(S)]$.

To apply this theorem, let us first assume that the variables X_i are discrete random variables over a finite domain. The universe U will then have one element for each possible instantiation of each variable X_i with a value equal to that instantiation. Then any joint instantiation of the variables X_1, \dots, X_n corresponds to a subset of U ; let D denote the corresponding distribution over subsets. Let f be the max function over the instantiated subset. Then $\mathbf{E}[\max_j X_j]$ is exactly equal to $\mathbf{E}_{S \sim D}[f(S)]$. As before, let \tilde{D} denote the distribution over subsets of U where each element is picked independently. Likewise, the random variables Y_1, \dots, Y_n define a distribution, say D' , over subsets of U . Note that under D' the memberships of elements of U in the instantiated subset are negatively correlated – for two elements that correspond to instantiations of the same variable, including one in the subset implies that the other is not included. This raises the expected maximum. In other words, $\mathbf{E}_{S \sim D'}[f(S)] \geq \mathbf{E}_{S \sim \tilde{D}}[f(S)]$. Therefore, we get $\mathbf{E}[\max_j X_j] = \mathbf{E}_{S \sim D}[f(S)] \leq (e/e-1) \mathbf{E}_{S \sim D'}[f(S)] = (e/e-1) \mathbf{E}[\max_j Y_j]$.

When the variables X_j are defined over a continuous but bounded domain, we can apply the above argument to an arbitrarily fine discretization of the variables. Our claim then follows from taking the limit as the granularity of the discretization goes to zero.

Finally, let us address the boundedness assumption. For some $\epsilon < 1/n^2$, let B be defined so that for all i , $\Pr[X_i > B] \leq \epsilon$. Then the contribution to the expected maximum from values above B is similar for the X s and the Y s: the probability that some variable X_i attains the maximum value $b > B$ is at most $\Pr[X_i = b]$ whereas the probability that the variable Y_i attains the maximum value $b > B$ is at least $(1 - \epsilon)^{n-1} \Pr[Y_i = b]$. Therefore, $\mathbf{E}[\max_j X_j] \leq (1 + o(\epsilon))(e/e-1) \mathbf{E}[\max_j Y_j]$. Taking the limit as ϵ goes to zero implies the theorem. ■

Comparing OPT and OPT_δ We now prove Lemma 2.1. The key intuition behind the lemma is that it can be viewed as the result of scaling both sides of the stochastic dominance relation of Lemma 2.9 up by a constant, and as we shall see, the monotone hazard rate condition is retained by the minimum among multiple draws from a probability distribution.

Lemma 2.1 *When the distributions of job sizes have monotone hazard rates the expected worst best and average best runtimes on δm machines are no more than $1/\delta^2$ times the expected worst best and average best runtimes respectively on m machines.*

Proof: We will show that the random variable $T_j[1:\delta m]$ is stochastically dominated by $\frac{1}{\delta} T_j[1:m]$. Then, the expected worst best runtime with δm machines is no more than $1/\delta$ times the expected worst best runtime with m machines. Likewise, the expected average best runtime with δm machines is no more than $1/\delta^2$ times the expected average best runtime with m machines. (The extra $1/\delta$ factor comes about because we average over δm machines for the former, versus over m machines for for the latter.)

Our desired stochastic dominance relation is precisely of the form given by Lemma 2.9. In particular, observe that taking a minimum among m draws is exactly the same as first splitting the m draws into $1/\delta$ groups, selecting the minimum from each group of δm draws, and then taking the minimum from this collection of $1/\delta$ values. Thus, we can see that $(1/\delta)T_j[1:m] = (1/\delta)T_j[1:\delta m][1:1/\delta]$, and so the claim follows immediately from Lemma 2.9 as long as the distribution of $T_j[1:\delta m]$ has a monotone hazard rate. We show in Claim 1 below that the first order statistic of i.i.d. monotone hazard rate distributions also has a monotone hazard rate. ■

Claim 1 *A distribution F has a monotone hazard rate if and only if the distribution of the minimum among k draws from F has a monotone hazard rate.*

Proof: Let F_k denote the cdf for minimum among n draws from F . Then we have $F_k(x) = 1 - (1 - F(x))^k$, and the corresponding $f_k(x) = k(1 - F(x))^{k-1}f(x)$. Thus the hazard rate function is:

$$h_k(x) = \frac{f_k(x)}{1 - F_k(x)} = \frac{k(1 - F(x))^{k-1}f(x)}{(1 - F(x))^k} = k \frac{f(x)}{1 - F(x)}.$$

This is precisely k times the hazard rate function $h(x)$, and therefore, $h_k(x)$ is monotone increasing if and only if $h(x)$ is. ■

6 Conclusions

Non-linear objectives coupled with multi-dimensional preferences present a significant challenge in mechanism design. Our work shows that this challenge can be overcome for the makespan objective when agents (machines) are a priori identical. This suggests a number of interesting directions for follow-up. Is the gap between the first-best and second-best solutions (i.e. the cost of incentive compatibility) still small when agents are not identical? Does knowledge of the prior help? Note that this question is meaningful even if we ignore computational efficiency. On the other hand, even if the gap is small, the optimal incentive compatible mechanism may be too complex to find or implement. In that case, can we approximate the optimal incentive compatible mechanism in polynomial time?

Similar questions can be asked for other non-linear objectives. One particularly interesting objective is max-min fairness, or in the context of scheduling, maximizing the running time of the least loaded machine. Unlike for makespan, in this case we cannot simply “discard” a machine (that is, schedule no jobs on it) without hurting the objective. This necessitates techniques different from the ones developed in this paper.

References

- [1] Shipra Agrawal, Yichuan Ding, Amin Saberi, and Yinyu Ye. Correlation robust stochastic optimization. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA ’10, pages 1087–1096, 2010.
- [2] Aaron Archer and Éva Tardos. Truthful mechanisms for one-parameter agents. In *FOCS*, pages 482–491, 2001.
- [3] Itai Ashlagi, Shahar Dobzinski, and Ron Lavi. An optimal lower bound for anonymous scheduling mechanisms. In *Proceedings of the 10th ACM conference on Electronic commerce*, EC ’09, pages 169–176, New York, NY, USA, 2009. ACM.
- [4] Jeremy Bulow and Paul Klemperer. Auctions vs negotiations. *American Economic Review*, 86(1):180–194, 1996.
- [5] S. Chawla, J. Hartline, D. Malec, and B. Sivan. Sequential posted pricing and multi-parameter mechanism design. In *STOC*, 2010.

- [6] Shuchi Chawla, Nicole Immorlica, and Brendan Lucier. On the limits of black-box reductions in mechanism design. In *STOC*, pages 435–448, 2012.
- [7] George Christodoulou, Elias Koutsoupas, and Annamária Kovács. Mechanism design for fractional scheduling on unrelated machines. In *ICALP*, pages 40–52, 2007.
- [8] George Christodoulou, Elias Koutsoupas, and Angelina Vidali. A lower bound for scheduling mechanisms. In *SODA*, pages 1163–1170, 2007.
- [9] E. Clarke. Multipart pricing of public goods. *Public Choice*, 11:17–33, 1971.
- [10] Edith Cohen, Michal Feldman, Amos Fiat, Haim Kaplan, and Svetlana Olonetsky. Envy-free makespan approximation: extended abstract. In *Proceedings of the 11th ACM conference on Electronic commerce*, EC ’10, pages 159–166, 2010.
- [11] Nikhil R. Devanur, Jason D. Hartline, Anna R. Karlin, and C. Thach Nguyen. Prior-independent multi-parameter mechanism design. In *WINE*, pages 122–133, 2011.
- [12] Peerapong Dhangwatnotai, Shahar Dobzinski, Shaddin Dughmi, and Tim Roughgarden. Truthful approximation schemes for single-parameter agents. In *FOCS*, pages 15–24, 2008.
- [13] Peerapong Dhangwatnotai, Tim Roughgarden, and Qiqi Yan. Revenue maximization with a single sample. In *ACM Conference on Electronic Commerce*, pages 129–138, 2010.
- [14] Amos Fiat and Ariel Levavi. Tight lower bounds on envy-free makespan approximation. *CoRR*, abs/1205.1786, 2012.
- [15] Theodore Francis Groves. Incentives in teams. *Econometrica*, 41(4):617–31, 1973.
- [16] J. Hartline and B. Lucier. Bayesian algorithmic mechanism design. In *STOC*, 2010.
- [17] J. Hartline and T. Roughgarden. Simple versus optimal mechanisms. In *ACM Conference on Electronic Commerce*, pages 225–234, 2009.
- [18] Elias Koutsoupas and Angelina Vidali. A lower bound of $1+\phi$ for truthful scheduling mechanisms. In *MFCS*, pages 454–464, 2007.
- [19] Ron Lavi and Chaitanya Swamy. Truthful mechanism design for multi-dimensional scheduling via cycle monotonicity. In *ACM Conference on Electronic Commerce*, pages 252–261, 2007.
- [20] Pinyan Lu. On 2-player randomized mechanisms for scheduling. In *WINE*, pages 30–41, 2009.
- [21] Pinyan Lu and Changyuan Yu. An improved randomized truthful mechanism for scheduling unrelated machines. In *STACS*, pages 527–538, 2008.
- [22] Pinyan Lu and Changyuan Yu. Randomized truthful mechanisms for scheduling unrelated machines. In *WINE*, pages 402–413, 2008.
- [23] Ahuva Mu’Alem. On multi-dimensional envy-free mechanisms. In *Proceedings of the 1st International Conference on Algorithmic Decision Theory*, ADT ’09, pages 120–131, 2009.

- [24] Ahuva Mu'alem and Michael Schapira. Setting lower bounds on truthfulness: extended abstract. In *SODA*, pages 1143–1152, 2007.
- [25] Noam Nisan and Amir Ronen. Algorithmic mechanism design (extended abstract). In *STOC*, pages 129–140, 1999.
- [26] Tim Roughgarden, Inbal Talgam-Cohen, and Qiqi Yan. Supply-limiting mechanisms. In *ACM Conference on Electronic Commerce*, pages 844–861, 2012.
- [27] W. Vickrey. Counterspeculation, auctions and competitive sealed tenders. *Journal of Finance*, pages 8–37, 1961.
- [28] Qiqi Yan. Mechanism design via correlation gap. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '11*, pages 710–719, 2011.